

AUTOMATIC SEGMENTATION IN DYNAMIC OUTDOOR ENVIRONMENTS

Amy Tabb*

Henry Medeiros

USDA-ARS-AFRS
Kearneysville, West Virginia, USA
amy.tabb@ars.usda.gov

Marquette University
Electrical and Computer Engineering
Milwaukee, Wisconsin, USA
henry.medeiros@marquette.edu

ABSTRACT

Segmentation in dynamic outdoor environments can be difficult when the illumination levels and other aspects of the scene cannot be controlled. In this paper, we describe a method that uses superpixels to determine low texture regions of the image that correspond to the background material, and then show how this information can be integrated with the color distribution of the image to compute optimal segmentation parameters for traditional binary segmentation as well as to produce silhouette probability maps. We show results of this algorithm in the context of an application for tree modeling.

Index Terms— segmentation, tree reconstruction, automation

1. INTRODUCTION

Segmentation is a key part of many automation contexts. In our application, we use segmented images in a pipeline to reconstruct the shape of leafless trees for automation applications [1]. Since data acquisition takes place outdoors, illumination conditions are not stable and may change rapidly and widely. Furthermore, the entire process is automated, and hundreds of images must be acquired per tree. Hence, the segmentation method must be robust and not require parameter tuning either over the course of acquiring images for one of tree or over an entire day of data acquisition.

The ability to robustly extract the silhouettes of objects of interest is generally an important step in the generation of three-dimensional models of complex objects such as trees. Existing silhouette extraction techniques based solely on thresholding and morphological characteristics of the object of interest, however, tend to generate unsatisfactory results, particularly with respect to segmentation. This problem, as most computer vision tasks, is further aggravated in dynamic

environments, which include situations such as drastically varying illumination conditions. Hence, we propose a novel method to segment an object (in this case a tree) from a low-texture background, which is robust to significant illumination changes. As explained in detail below, the proposed method does not compute traditional binary silhouettes. Instead, it generates more general silhouette probability maps. These maps can then be trivially thresholded to produce traditional binary segmentations, if necessary.

There is a great deal of related work on segmentation in dynamic environments for the purposes of foreground detection as recently reviewed by Bouwmans [2]. Traditionally, in foreground detection, the assumption is that the background can be modeled because the cameras observing a scene are not moving. The background image hence remains relatively static and objects that move with respect to the camera are considered part of the foreground. A popular approach for foreground, or motion, detection is that of Stauffer and Grimson [3], in which each image pixel is modeled as a mixture of Gaussians. Various extensions of [3], from a hierarchical approach for real-time execution [4] to models that consider non-Gaussian distributions [5], have been explored as well for the context of relatively static backgrounds. In the scenarios under consideration in this work, however, the background may change drastically, so motion detection approaches are not applicable.

In the agricultural context, many applications require the segmentation of plants from soil with a moving camera; this problem has been recently surveyed by Hamuda *et al.* [6]. Concerning applications of tree segmentation, Byrne and Singh [7] use co-occurrence statistics to oversegment images into tree versus non-target tree regions for use in autonomous diameter measurement for a automated forestry application. In a similar application, Ali [8] uses a combination of color and texture features fed into an artificial neural network and k-nearest neighbor classifiers to perform classification of pixels into tree and non-tree classes. We build on these works to propose a robust and fast method that performs extremely well in challenging real-world conditions.

*Mention of trade names or commercial products in this publication is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture. USDA is an equal opportunity provider and employer. A. Tabb acknowledges the support of US National Science Foundation grant number IOS-1339211.

2. METHOD DESCRIPTION

We assume a low-texture background object is present in each image, and we model the hue component of this background object according to a normal distribution $\mathcal{N}(\mu, \sigma^2)$. In the following steps, we show how we estimate this distribution and then use it to assign probabilities for each pixel in the image. We also assume that the object of interest is positioned between the background object and the camera (see Figure 1a). Regions that extend beyond the background object are truncated. Algorithm 1 shows an overview of the proposed approach. Each step of the algorithm is explained in detail in the following subsections. While the first two steps of the algorithm are independent and can be performed in parallel, the remaining steps depend upon one another and hence need to be performed in order. The sequence of steps is illustrated in Figure 1.

Algorithm 1 Proposed silhouette map generation approach

Input: Image in hue-saturation-value (HSV) color space

Output: Segmented image

- 1: Compute the set of superpixels \mathbb{S} using the SEEDS method [9] and find the subset $\mathbb{R} \subset \mathbb{S}$ of low-texture superpixels.
 - 2: Generate a binary image T , by thresholding the hue channel using Otsu’s algorithm.
 - 3: Determine $\mathcal{N}(\mu, \sigma^2)$, the distribution of the background, based on \mathbb{R} and T .
 - 4: Assign labels to individual output pixels according to $\mathcal{N}(\mu, \sigma^2)$.
 - 5: Create a mask to eliminate regions outside of the background object.
-

2.1. Step 1: Computation of low-texture superpixels

The first step of our approach consists of converting the image to the HSV color space and partitioning it into superpixels. We have chosen to compute the superpixels using the superpixels extracted via energy-driven sampling (SEEDS) method proposed in [9] and implemented in OpenCV [10]. In this superpixel approach, the image is divided into a grid pattern, which serves as initial superpixel assignment. The superpixel assignments are refined by iteratively modifying their boundaries.

We set the parameters of the SEEDS algorithm such that low texture regions have superpixels whose shape is unchanged from the initial grid assignment. That is, let \mathbb{S} be the set of superpixels generated by SEEDS. Then there is a set of superpixels $\mathbb{R} \subset \mathbb{S}$, which are rectangular in shape. These are the unchanged superpixels which correspond to low-texture regions. \mathbb{R} corresponds to the smallest block size of the SEEDS algorithm, which can be determined based on the size of the images and the parameters chosen for SEEDS.

Figures 1a and 1b show the original RGB image and the hue channel of its corresponding HSV representation. Figure 1c then shows the superpixels generated according to our proposed procedure. As the image shows, most superpixels on the background object are rectangular in shape.

2.2. Step 2: Generation of thresholded hue image using Otsu’s algorithm

We then generate the binary image T by thresholding the hue channel of the HSV image using Otsu’s algorithm [11]. In the binary image, pixels with value above the threshold value are black and the remaining pixels are white. As explained in the next section, the image T is used to generate hypotheses of low-texture regions in the image, since the color of the low-texture background object is relatively constant. In our application, the blue background object has a lower hue value than other common colors in the images such as brown, gray, or green, which facilitates the application of the proposed approach. We are not limited to a single background color though. As long as the hue value of the majority of the pixels in the background differ from those of the foreground object, T can be generated using multi-level thresholding algorithms [12].

2.3. Step 3: Estimation of the distribution of the background

The first two steps consisted of somewhat unreliable detectors for the blue background. The superpixel approach in step one finds low texture regions, while T found in step two indicates regions likely to be the background judging by relative hue as compared to the rest of the image. We now combine the information from these two steps to generate a more robust background detector.

We determine regions where T overlaps superpixels in \mathbb{R} using the procedure summarized in Algorithm 2. Briefly, the algorithm iterates through the superpixels in \mathbb{R} . If the percentage of white pixels in the corresponding area of the thresholded image T exceeds a value ζ , then the superpixel is added to a set \mathbb{B} . The set \mathbb{B} hence consists of all the superpixels which belong to a low-texture region as determined by the SEEDS algorithm *and* by its constant color.

All of the pixel locations in the set of superpixels \mathbb{B} are then used to estimate the probability distribution of the background pixels. Let h_i be the value of the i th pixel in the hue image. We assume the pixel intensity is normally distributed according to $h_i \sim \mathcal{N}(\mu, \sigma^2)$, that is, its probability density is given by $p(h_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(h_i - \mu)^2}{2\sigma^2}}$. The parameters of the distribution, μ and σ , are obtained from the sample mean and the sample variance of the pixels in \mathbb{B} . To avoid numerical issues, we set the minimum value of the sample variance to an empirically determined value of $\sigma_{min} = 4$.

Algorithm 2 Determination of background pixels

Input: Set \mathbb{R} of low-texture superpixels and thresholded hue image T

Output: Set \mathbb{B} of background superpixels

- 1: $\mathbb{B} = \{\}$
 - 2: **for** each superpixel $r_i \in \mathbb{R}$ **do**
 - 3: $t_i =$ number of white pixels in T for the region of r_i .
 - 4: **if** $t_i/\text{area}(r_i) > \zeta$ **then**
 - 5: $\mathbb{B} = \mathbb{B} \cup r_i$
-

2.4. Step 4: Pixel label assignment based on the background distribution

The hue image is compared to the distribution $\mathcal{N}(\mu, \sigma^2)$ to generate an image of pixel labels L as follows. Let F_i be an indicator random variable that is 1 if the i th pixel in the hue image belongs to the object of interest or zero otherwise. The random variable F_i conditioned on h_i is Bernoulli distributed according to $p(F_i|h_i) = \phi$ where the parameter $\phi = 1 - \sigma\sqrt{(2\pi)p(h_i)}$. The label of the i th pixel of the image L is then given by $255 \times p(F_i|h_i)$.

2.5. Step 5: Create mask to eliminate regions outside the background object

The final step creates a mask image M , in which regions including the object of interest, in this case the tree, and the background object, are labeled white and all other regions are labeled black. This mask is generated over the course of a few steps. Initially, all the pixels in \mathbb{B} as well as pixels i where $p(h_i) \geq p_m$ are labeled white in M . Small connected components with area smaller than a threshold of ϵ_a are removed from M . We then perform successive dilation operations on M until there is only one connected component. The number of dilation operations required is denoted d . The contour, convex hull and convexity defects are computed from the connected component. Convexity defects with depth (i.e., the distance between the furthest point in the convexity defect to the convex hull) exceeding a threshold (ϵ_d) are then filled, as well as all interior contours. Finally, the mask is restored to its original dimensions by eroding it d times.

The mask is applied to the pixel label image L so that if the i th pixel of M is 0, then this pixel is outside the region of interest and the corresponding pixel in L is labeled accordingly. In this application, we label those pixels 0 and the pixels inside the region of interest are labeled $255 \times p(F_i|h_i)$.

3. RESULTS

We evaluate our proposed approach in images of six trees collected outdoors under varying illumination conditions as shown in Figure 2. In our experiments, the threshold parameter $\zeta = 0.8$, and the mask generation parameters are

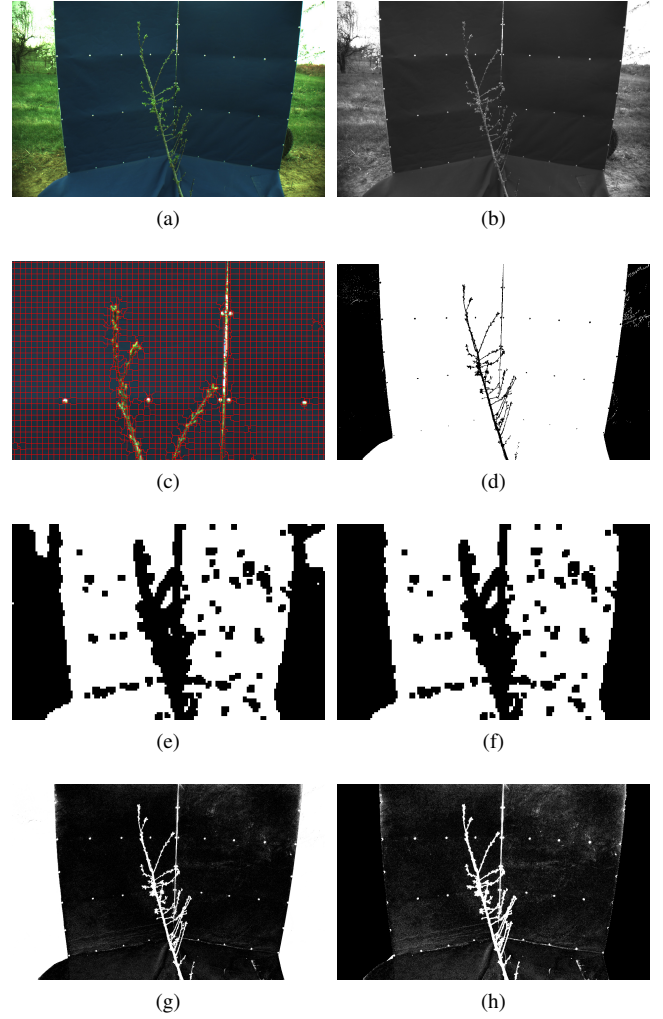


Fig. 1. [Best viewed in color] (a) Original RGB image showing the object of interest (tree) in front of the background object. (b) Hue channel. (c) Close-up of portion of the original image (top portion of branch) with superpixels overlaid in red. (d) Threshold result from step 2 (T). (e) The set of superpixels \mathbb{R} is shown in white, indicating low texture regions. (f) Set of superpixels \mathbb{B} , where white pixels indicate locations where the distribution $\mathcal{N}(\mu, \sigma^2)$ is estimated in the hue image. (g) L after step 4. (h) L after application of the mask in step 5.

$p_m = 1e^{-4}$, $\epsilon_a = 500 = \epsilon_d = 500$. The resolution of the images is 1900×1200 and the SEEDS parameters that guarantee that the low-texture areas remain unchanged are: number iterations = 10, number histogram bins = 2, number superpixels = 16,000, number levels = 1, prior = 1, and double step = false.

In order to quantify the results of our method, we binarize the silhouette probability maps using a fixed threshold of 240, which corresponds to pixels with probability of belonging to the object of interest higher than 94%. We then com-

Table 1. Quantitative results comparing the baseline method (B/L) with our proposed approach in terms of false positive rate (FPR), false negative rate (FNR) and the F-score. Best results are shown in boldface.

	FPR		FNR		F-score	
	B/L	Ours	B/L	Ours	B/L	Ours
1	53.68	2.97	0.77	0.71	0.632	0.982
2	0.21	0.50	0.40	0.14	0.997	0.997
3	0.43	0.76	0.65	0.33	0.995	0.995
4	0.27	0.21	0.21	0.14	0.998	0.998
5	0.13	0.83	1.00	0.55	0.994	0.993
6	0.25	0.61	0.32	0.07	0.997	0.997
Mean	9.16	0.98	0.56	0.32	0.936	0.994
Median	0.26	0.69	0.53	0.24	0.996	0.996

pare the binarized image with a baseline approach that simply binarizes the hue channel of the original images using Otsu’s algorithm. Since the baseline approach cannot automatically detect pixels outside the background object, we also provide it the bounding box within which the object of interest can be found. Hence, although simple, the baseline approach proved effective under most relatively simple conditions. It should be noted that not only does our method have no knowledge of the target bounding box but also that simply thresholding the probabilities is a rather naïve approach to utilize the information provided by our algorithm. Our reconstruction methods, for example, directly use these probability maps without resorting to binarization [1], [13].

Table 1 summarizes the performance of both approaches in terms of the false positive rate (FPR), false negative rate (FNR) and the F_1 score. The ground truth data was generated by hand-labeling the tree versus non-tree pixels in the six images with an image editor. As the table shows, our method outperforms the baseline method in terms of FNR in every image and also in FPR in most images, particularly in more challenging scenarios such as the one shown in Figure 2a.

4. CONCLUSION

In this paper we proposed a method to perform automatic segmentation of objects of interest in dynamic outdoor conditions. We are interested in automation scenarios in which an object of interest must be segmented from a low-texture background such as in tree reconstruction. Our method estimates the probability distribution of the low-texture background by fusing information from its color distribution and from superpixels extracted from the background. As a result, it is particularly robust to substantial variations in illumination conditions, significantly outperforming a baseline approach in challenging illumination conditions.

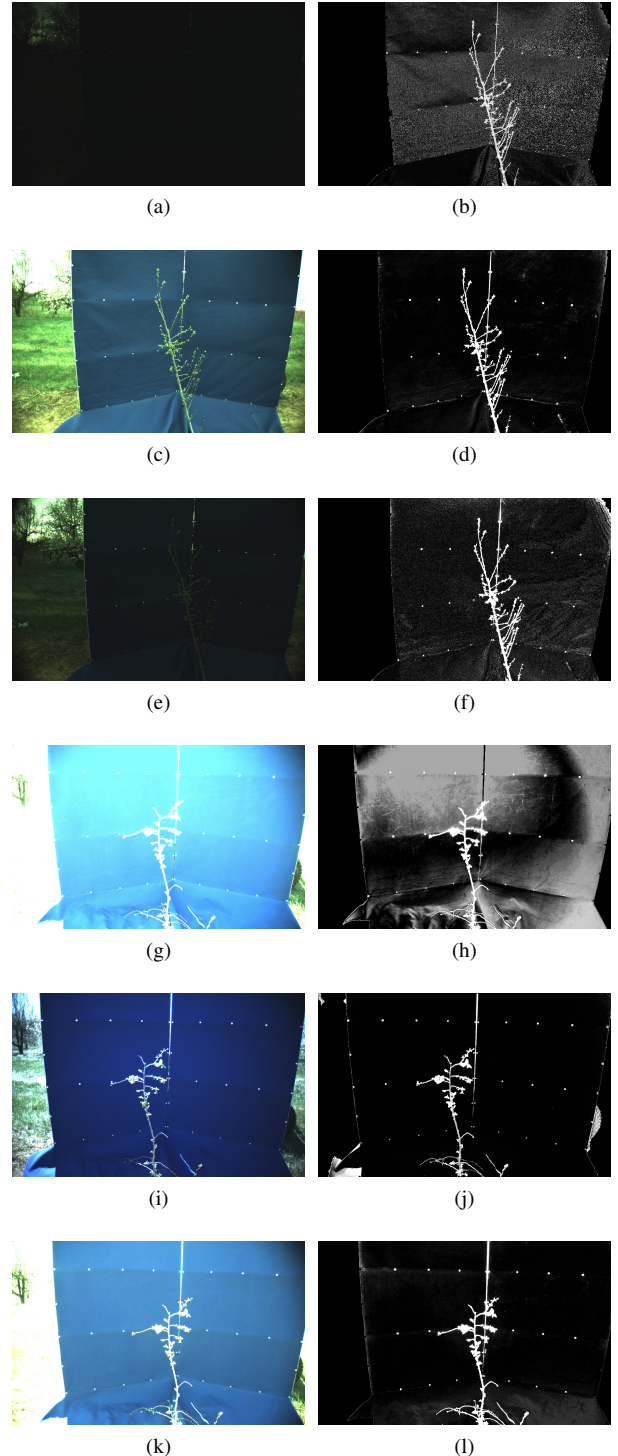


Fig. 2. [Best viewed in color] Images of trees in front of a low-texture blue background with varying illumination levels (left column) and the corresponding segmentation probability maps obtained using the proposed approach (right column).

5. REFERENCES

- [1] A. Tabb, “Shape from silhouette probability maps: reconstruction of thin objects in the presence of silhouette extraction and calibration error,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013.
- [2] Thierry Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Computer Science Review*, vol. 1112, pp. 31 – 66, 2014.
- [3] Chris Stauffer and W Eric L Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*. IEEE, 1999, vol. 2.
- [4] J. Park, A. Tabb, and A. C. Kak, “Hierarchical data structure for real-time background subtraction,” in *2006 International Conference on Image Processing*, Oct 2006, pp. 1849–1852.
- [5] Antoni B. Chan, Vijay Mahadevan, and Nuno Vasconcelos, “Generalized stauffer–grimson background subtraction for dynamic scenes,” *Machine Vision and Applications*, vol. 22, no. 5, pp. 751–766, 2011.
- [6] Esmael Hamuda, Martin Glavin, and Edward Jones, “A survey of image processing techniques for plant extraction and segmentation in the field,” *Computers and Electronics in Agriculture*, vol. 125, pp. 184 – 199, 2016.
- [7] Jeffrey Byrne and Sanjiv Singh, *Precise image segmentation for forest inventory*, Carnegie Mellon University, The Robotics Institute, 1998.
- [8] Wajid Ali, “Tree detection using color, and texture cues for autonomous navigation in forest environment,” *Umeå University*, 2006.
- [9] Michael Van den Bergh, Xavier Boix, Gemma Roig, and Luc Van Gool, “Seeds: Superpixels extracted via energy-driven sampling,” *International Journal of Computer Vision*, vol. 111, no. 3, pp. 298–314, 2015.
- [10] “Opencv,” <http://opencv.org/>, version 3.0.
- [11] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.
- [12] Deng-Yuan Huang and Chia-Hung Wang, “Optimal multi-level thresholding using a two-stage otsu optimization approach,” *Pattern Recognition Letters*, vol. 30, no. 3, pp. 275 – 284, 2009.
- [13] Amy Tabb, *Shape from inconsistent silhouette: Reconstruction of objects in the presence of segmentation and camera calibration error*, Ph.D. thesis, Purdue University, 2014.